

aahh 10 11

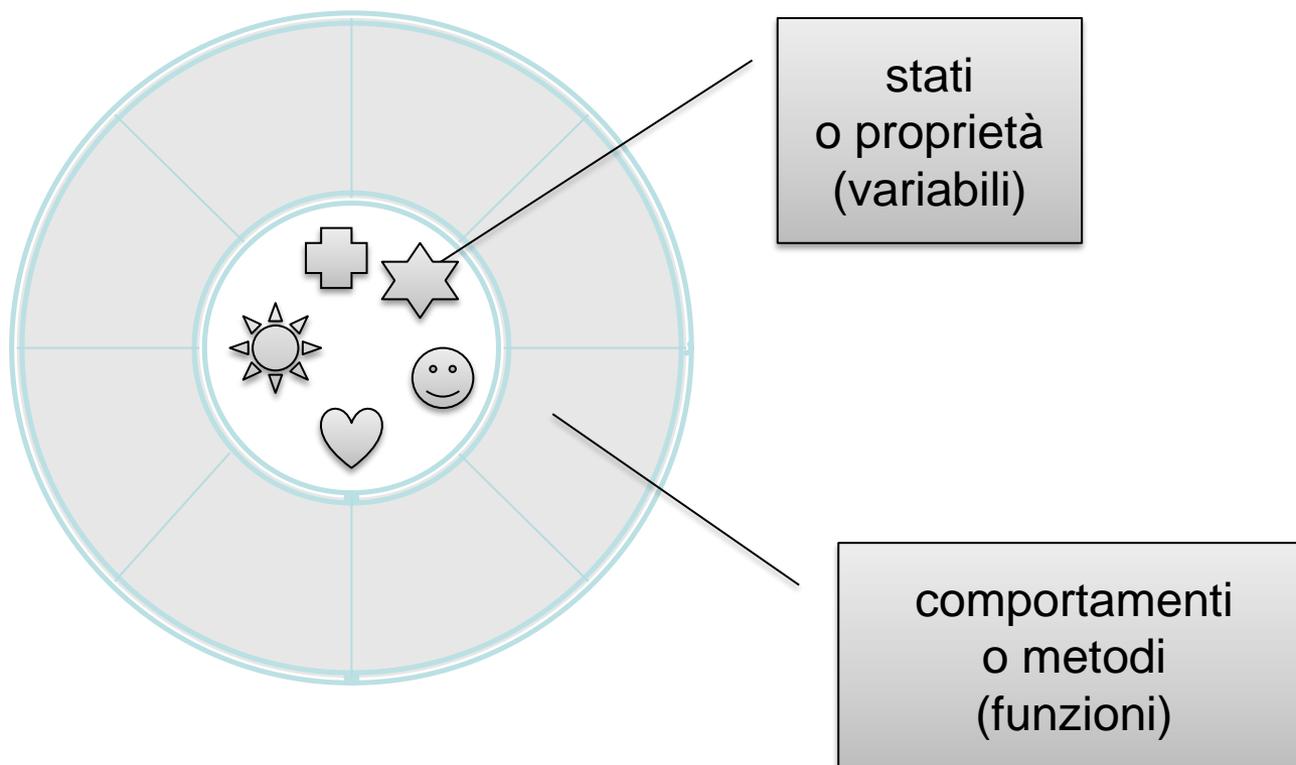
ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE

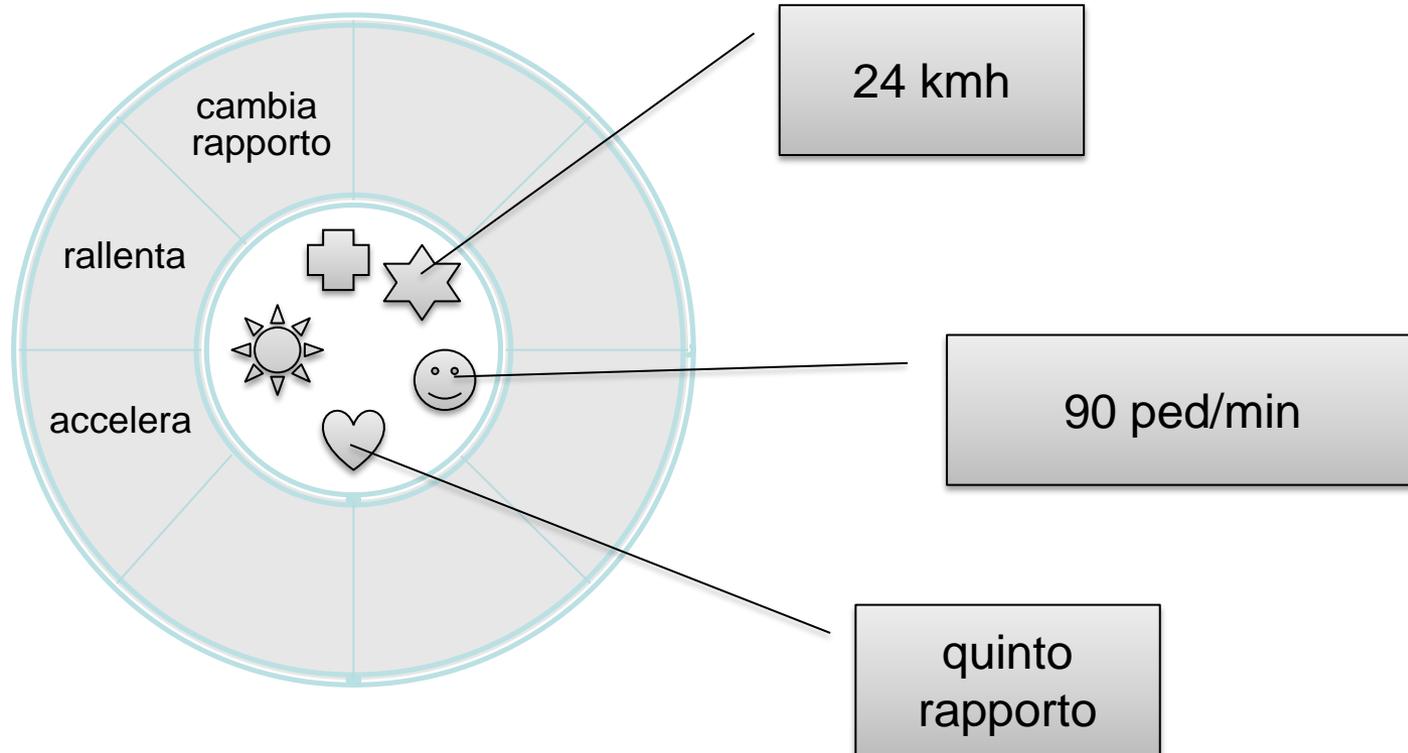


LE CLASSI

Oggetto software



Bicicletta



ISTANZE E MEMBRI DI CLASSE

- Nel mondo reale, spesso troverete molti singoli oggetti tutti dello stesso tipo. Ci possono essere migliaia di biciclette esistenti, tutti della stessa marca e modello.
- Ogni bicicletta è stata costruita sulla base dello stesso progetto e quindi contiene gli stessi componenti.
- Una classe definisce un modello per un tipo di oggetto. Tutte le caratteristiche e i comportamenti che appartengono a una classe sono detti *membri* di tale classe. In particolare, le caratteristiche (nell'esempio del gatto, il nome, l'età e il colore) sono dette *proprietà* della classe e sono rappresentate da variabili, mentre i comportamenti (mangiare, dormire) sono detti *metodi* della classe e sono rappresentati da funzioni.

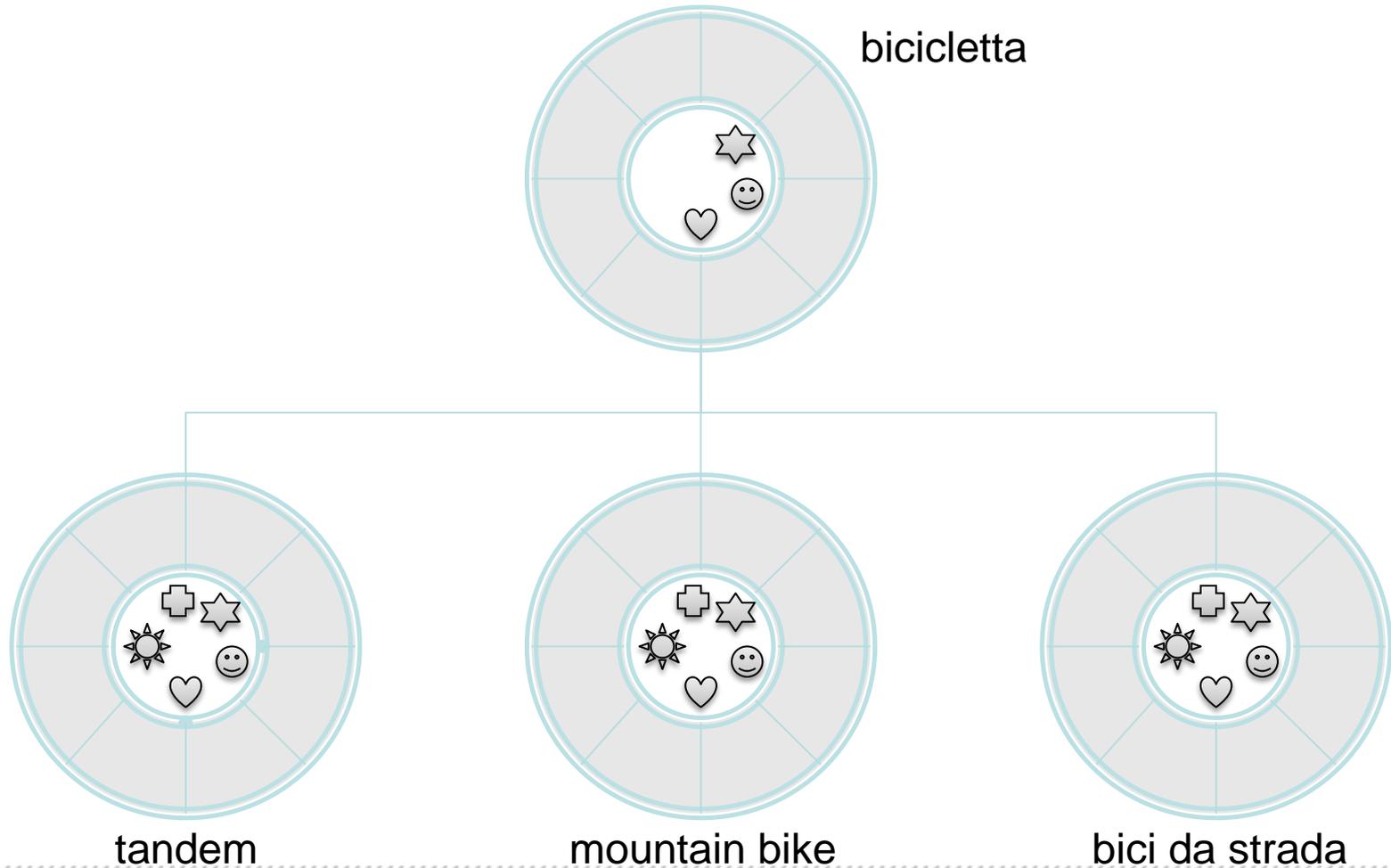
EREDITARIETÀ

- Mountain bike, biciclette da strada e tandem, condividono alcune caratteristiche (velocità, cadenza della pedalata, rapporto).
- Ma ciascuno definisce anche elementi aggiuntivi che le rendono differenti:
 - i tandem hanno due posti e due set di manubri,
 - le bici da strada hanno manubrio da corsa;
 - le mountain-bikes hanno una corona aggiuntiva che consente loro rapporti di trasmissione inferiori.
- La programmazione orientata agli oggetti consente di definire classi che *ereditano* gli stati e i comportamenti di altre classi. In questo esempio, bicicletta può essere una *superclasse* e MountainBike, Bici da strada, e Tandem delle sottoclassi.

EREDITARIETÀ

- Uno dei vantaggi principali della programmazione orientata agli oggetti è rappresentato dalla creazione di **sottoclassi** (tramite estensione di una classe); una **sottoclasse** eredita tutte le proprietà e i metodi della rispettiva classe.
- La sottoclasse definisce generalmente ulteriori metodi e proprietà o sostituisce metodi o proprietà definiti nella superclasse.
- Le sottoclassi possono inoltre sostituire i metodi definiti in una superclasse, ovvero fornire definizioni proprie per tali metodi.

EREDITARIETÀ



INTERFACCIE

- Gli oggetti definiscono la loro interazione con il mondo esterno attraverso i metodi che espongono.
- I metodi costituiscono l'interfaccia con il mondo esterno, nello stesso modo in cui i pulsanti sulla parte anteriore del televisore, per esempio, sono l'interfaccia tra l'utente e l'elettronica che contiene involucro di plastica.
- Nella nostra classe bicicletta, l'interfaccia potrebbe essere per esempio l'elenco dei metodi che devo definire obbligatoriamente per creare una sottoclasse di bicicletta.

Package

- Un pacchetto è un namespace che organizza una serie di classi e interfacce correlate. Si può pensare di pacchetti come a diverse cartelle del computer. Poiché il software scritto in linguaggio ActionScript può essere composto da centinaia di singole classi è importante mantenere le classi organizzate in pacchetti.
- ActionScript fornisce una biblioteca (un insieme di pacchetti) che contengono le classi con cui abbiamo bisogno per gestire gli elementi. Questa libreria è noto come "Application Programming Interface", o "API".

CLASSI E TIPI

- Quando abbiamo parlato dei tipi di dati abbiamo visti che esistono **tipi di dati primitivi** e **tipi di dati derivati** o **complessi**.
- In un linguaggio orientato agli oggetti tutti i tipi di dati sono classi. Definiscono cioè non solo come un certo tipo di dato viene memorizzato nel computer, ma anche i metodi con cui posso interagire con esso.
- Creando una nuova classe il programmatore crea un nuovo **tipo di dati complesso**, un **modello**, cioè, di come posso gestire un determinato insieme di informazioni e di come posso interagire con esso.

aahh 10 11

ACCADEMIA DI BELLE ARTI DI URBINO

SISTEMI INTERATTIVI DUE

CREAZIONE DI CLASSI

LE CLASSI

- Nella programmazione orientata agli oggetti una classe definisce una *categoria di oggetti*.
- Le classi in pratica sono nuovi tipi di dati che il programmatore può creare per definire un nuovo tipo di oggetto. Una classe descrive le *proprietà (dati)* e i *metodi (comportamenti)* di un oggetto in modo molto simile a come un progetto di design descrive le caratteristiche di un oggetto, o a come una classificazione botanica descrive le caratteristiche di una pianta.
- Le proprietà (variabili definite all'interno di una classe) e i metodi (funzioni definite all'interno di una classe) di una classe sono detti *membri* della classe.
- In generale per utilizzare le proprietà e i metodi definiti da una classe, è necessario creare prima un'istanza di tale classe. La relazione tra un'istanza e la relativa classe è simile a quella che intercorre tra un oggetto reale e il relativo progetto del designer o tra una pianta reale e la sua classificazione.

LA CLASSE BIKE

Rapporto
usato



Diametro
ruota



Velocità



Cadenza

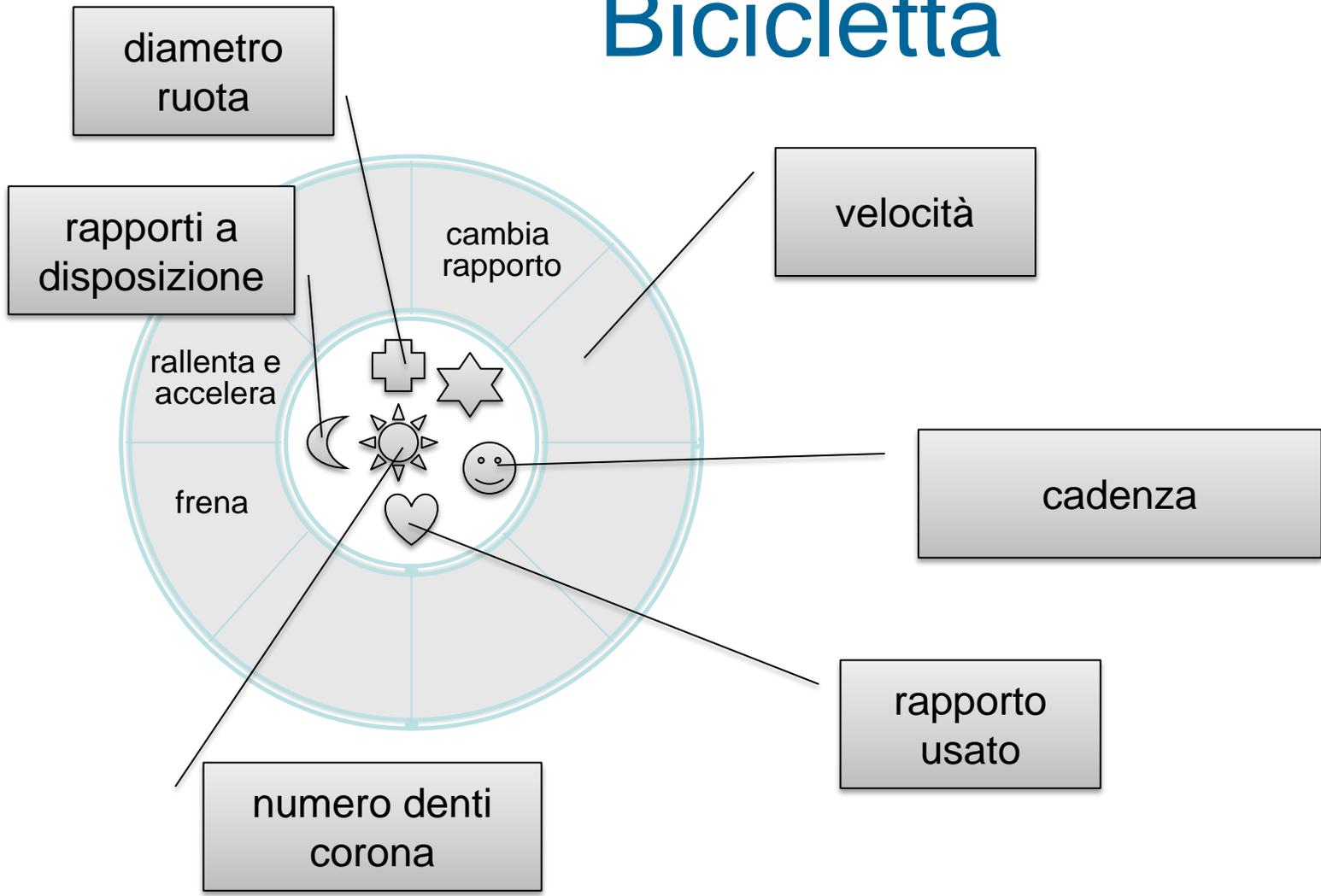


numero
denti

numero
denti

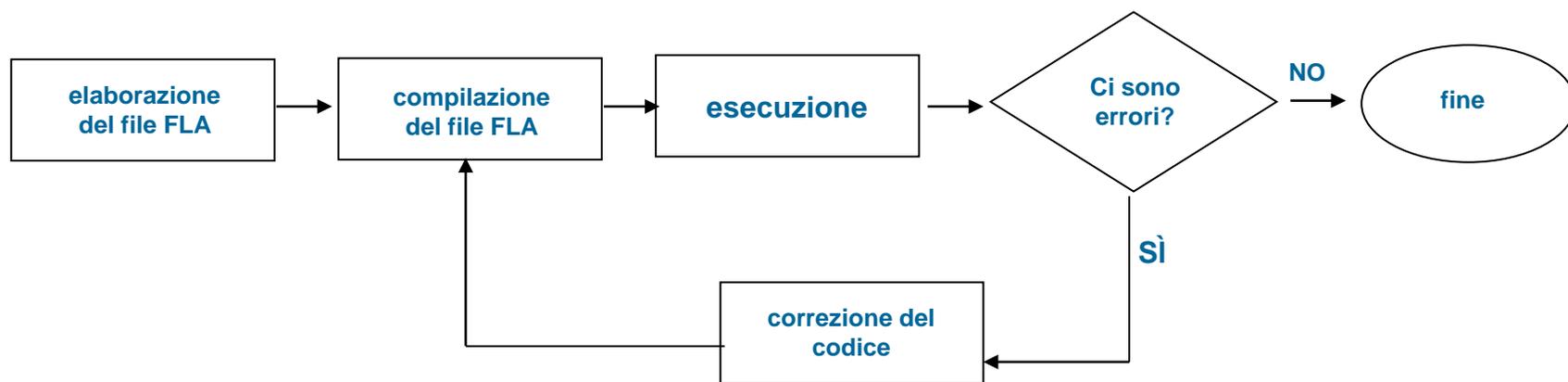
diametro
ruota

Bicicletta



IL PROCESSO

- Il processo di realizzazione di un progetto flash è strutturata in un flusso simile a questo:



- Con la programmazione con le classi introduciamo un nuovo tipo di file: il file **Action Script (.as)** ma il processo rimane lo stesso.
- Se il progetto comprende anche dei file .as il compilatore li unirà ai dati contenuti nel .fla e creerà un unico file shockwave (filmato flash compilato e compresso).

I FILE DI CLASSE

- Il file **.as** è un normale file di testo (tipo blocco note) che contiene codice Action Script.
- Una classe in *ActionScript* viene sempre definita in un file esterno (un normale file di testo con estensione **.as**) che ha lo stesso nome della classe e che viene chiamato *file di classe*.
- Quando un filmato flash viene compilato (utilizzando Controllo > Prova filmato o File > Pubblica) per generare il file **.swf**, il codice contenuto nei file di classe necessari viene compilato e aggiunto al file **.swf**.

I FILE DI CLASSE

- Quando il compilatore trova che nel filmato da compilare viene utilizzata una classe **DEVE** trovare il file che contiene il codice relativo a quella classe:
- Il file di classe deve avere esattamente lo stesso nome della classe (case sensitive).
- Il compilatore deve sapere in che cartelle cercare.

I FILE DI CLASSE

1. Il compilatore in primo luogo inizierà la sua ricerca dalla cartella in cui è stato salvato il file .fla.
2. Esiste un elenco globale di cartelle che contengono classi che si può modificare andando in: Modifica>Preferenze>ActionScript e scegliendo il bottone “Impostazioni Action Script 3”

I PACCHETTI

- In ActionScript 3 è fondamentale capire il concetto di package.
- Con l'aggiunta del comando **package** (obbligatorio) per ogni classe viene definito un **package** di appartenenza che corrisponde a come i file di classe sono organizzati su disco
- Supponiamo che il mio progetto Bicicletta fla si collocato in **D:\ProgettiFlash\Bicicletta**

I PACCHETTI

```
//package aninimo
package {
  public class Bike{
    ...
  }
}
```

- Il file di classe si chiamerà **Bike.as** e risiederà nella cartella del progetto:
D:\progettiFlash\Rosa\Bike.as

I PACCHETTI

```
//package con nome
package classiProgetto{
    public class Rosa {
        ...
    }
}
```

Il file di classe si chiamerà **Bike.as** e risiederà nella cartella classiProgetto:

D:\progettiFlash\Rosa\classiProgetto\Bike.as

I PACCHETTI

```
//package con nome  
package classiProgetto.uno{  
    public class Rosa {  
        ...  
    }  
}
```

Il file di classe si chiamerà **Bike.as** e risiederà nella cartella classiProgetto\uno:

D:\progettiFlash\Rosa\classiProgetto\uno\Bike.as

LA CLASSE BIKE

```
package
```

```
{
```

```
    public class Bike
```

```
    {
```

```
        //definizione della classe
```

```
        . . .
```

```
        . . .
```

```
        . . .
```

```
    }
```

```
}
```

LA CLASSE ROSA

```
public class Bike{  
    // Proprietà della classe  
    private var rapporti:Array = [24,  
        22, 20, 18, 16, 14];  
    private var corona:uint = 48;  
    private var diametroRuota:Number;  
  
    public var cadenza:Number;  
    public var statoCambio:uint;  
  
}
```

LA CLASSE ROSA

```
public class Bike{  
    ..  
    /* Metodo "Constructor" : viene chiamato  
    quando si crea una istanza della classe  
    */  
    public function Bike(misura:uint):void  
    {  
        diametroRuota = misura * 2.54001;  
        cadenza = 0;  
        statoCambio = 2;  
    }  
    ..  
}
```

LA CLASSE ROSA

```
public class Bike{  
  ..  
  // Metodo per cambiare l Cadenza  
  public function cambiaCadenza (diff:Number) :void  
  {  
    cadenza = cadenza + diff;  
    if (cadenza < 0)  
    {  
      cadenza = 0;  
    }  
  }  
}
```

LA CLASSE ROSA

```
public class Bike{  
  ..  
  // Metodo per ricavare la velocità  
  public function speed ():Number  
  {  
    var passo = diametroRuota * Math.PI *  
      corona / rapporti[statoCambio];  
    var km_ora = passo * cadenza * 60 /100000;  
    return km_ora;  
  }  
}
```

PROVARE UNA CLASSE

- Per creare e usare una classe è necessario:
 - Definizione di una classe in un file di classe *ActionScript* esterno.
 - Salvataggio del file di classe nella directory specificata per il percorso della classe (o nel percorso in cui Flash cerca le classi) oppure nella stessa directory del file FLA dell'applicazione.
 - Creazione di un'istanza della classe in un altro script, ossia un documento FLA o un file di script esterno, oppure tramite creazione di una sottoclasse basata sulla classe originale.

USARE UNA CLASSE

- Per creare un'istanza di una classe *ActionScript*, si utilizza l'operatore **new** per richiamare la funzione di costruzione della classe. Tale funzione ha sempre lo stesso nome della classe e restituisce un'istanza della classe che generalmente viene assegnata a una variabile.

```
var bici:Bike = new Byke();
```

- Usando l'operatore punto (.) si accede al valore di una proprietà o a un metodo di un'istanza.

```
Bici.cambiaCadenza(-2);
```

GLI ATTRIBUTI DI CONTROLLO DI ACCESSO

- Gli attributi di controllo di accesso determinano la visibilità o scopo di classi, proprietà e metodi.
- La sintassi è la seguente
- <attributo> **class**
- <attributo> **var**
- <attributo> **function**

GLI ATTRIBUTI DI CONTROLLO DI ACCESSO

public	Visibilità completa
internal	Visibilità limitata alle classi che si trovano nello stesso package
private	Visibilità limitata alla sola classe di appartenenza
protected	Visibilità limitata alla classe di appartenenza e alle sottoclassi

METODI E PROPRIETÀ STATICI

- La parola chiave *static* specifica che una variabile o una funzione viene creata solo una volta per ogni classe anziché in ogni oggetto basato sulla classe. È possibile accedere a un membro di classe statico senza creare un'istanza della classe. I metodi e le proprietà statici possono essere sia pubblici che privati.
- In ActionScript ci sono classi predefinite che hanno solo metodi e proprietà statiche.